

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**  
**дисциплины «Алгоритмизация»**

Выполнил:  
Епифанов Алексей Александрович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных систем  
», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Написал программу подсчета инверсий в массиве за время  $O(n \cdot \log(n))$ :

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import random as rnd
5  from collections import deque
6
7
8  def create(length, max_value):
9      unique_numbers = list(range(max_value))
10     rnd.shuffle(unique_numbers)
11     return unique_numbers[:length]
12
13 def merge_sort(arr):
14     if len(arr) <= 1:
15         return arr, 0
16     else:
17         middle = len(arr) // 2
18         left, inv_left = merge_sort(arr[:middle])
19         right, inv_right = merge_sort(arr[middle:])
20         merged, inv_merge = merge(deque(left), deque(right))
21         return merged, inv_left + inv_right + inv_merge
22
23 def merge(left, right):
24     merged = []
25     inv_count = 0
26     while left and right:
27         if left[0] <= right[0]:
28             merged.append(left.popleft())
29         else:
30             merged.append(right.popleft())
31             inv_count += len(left)
32     merged.extend(left)
33     merged.extend(right)
34     return merged, inv_count
35
36
37 if __name__ == '__main__':
38     array = create(5, 100)
39     # array = [5, 4, 3, 2, 1, 6, 7, 8]
40     print("Array =", array)
41     _, count = merge_sort(array)
42     print("Количество инверсий в массиве =", count)
43
```

Рисунок 1. Код программы

```
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [44, 61, 89, 85, 15]
Количество инверсий в массиве = 5
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [30, 29, 5, 27, 98]
Количество инверсий в массиве = 5
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [15, 23, 40, 86, 18]
Количество инверсий в массиве = 3
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [33, 12, 0, 29, 49]
Количество инверсий в массиве = 4
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [67, 38, 60, 59, 53]
Количество инверсий в массиве = 7
aleksejeripfanov@MacBook-Pro program % ./main.py
Array = [4, 54, 45, 91, 53]
Количество инверсий в массиве = 3
aleksejeripfanov@MacBook-Pro program % |
```

Рисунок 2. Результат работы программы

В ходе выполнения лабораторной работы было исследовано, каким образом лучше подсчитывать инверсии в массиве. Выяснилось, что интуитивно понятный метод работает за квадратичное время, поэтому необходимо использовать более быстрые алгоритмы, например метод сортировки слиянием с подсчетом инверсий, благодаря которому удалось достичь времени  $O(n \cdot \log(n))$ .